National College of Ireland

Project Submission Sheet – 2021/2022

Student Name:	Aditya Pramod Shinde		
Student ID:	20178883		
Programme:	MSCDAD_B	**Year:**	2021-2022
Module:	Data Mining And Machine Learning		
Lecturer:	Anu Sahni		
Submission Due Date:	21 – 12 -2021		
Project Title:	Exploring Machine Learning Models for classification and prediction on Football Data		
Word Count:	4318 Words		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	Aditya Pramod Shinde
Date:	21 – 12 -2021

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Exploring Machine Learning Models for classification and prediction on Football Data

Aditya Pramod Shinde
MSc. in Data Analytics
National College Of Ireland
Dublin, Ireland
x20178883@student.ncirl.ie

***Abstract*—This report aims to utilize KDD (Knowledge Discovery in Databases) methodology to apply various machine learning algorithms to achieve the predictions and classification tasks. The tasks performed are goals prediction for football matches using regression and classification of football match outcomes. Before applying various machine learning models data exploration, data cleaning, and transformation were performed for all the datasets. All the utilized models were evaluated using various parameters. The classification task was performed using Random Forest and the XGBoost algorithm. The overall accuracy of 61% was achieved for our classification task. For our prediction task multiple linear regression, the Lasso regression, and the Ridge regression were used. The parameters of our models were hyper tuned to obtain the best outcomes from the models. For evaluation of these models, RMSE (root mean squared error) and MAPE (mean absolute percentage error) were utilized. The conclusions were subsequently formed and comprehensively reported.**

***Index Terms*—multiple regression, credit debt, prediction**

I. Introduction

Sports have always been an integral part of human society and even before any sporting event begins people love to predict the outcome of the event. This not only makes the game more interesting but also allows people to wager and make some profit out of it. Football is considered as one of the most popular and the most betted upon sports in the world. It is a high-stakes game where billions are invested by the teams and sponsors in the sport. There has always been a neck-to-neck competition among the clubs to gain an edge over their opponents in any way possible. Football prediction can be a vital insight for each team to evaluate their performance and formulate strategies for their future games. The game prediction is also popular around the wagering community which studies various statistics and probabilities to gain an upper hand and make a profit. Although, football match prediction is easier said than done. Various factors come into consideration and play when carrying out such tasks. The slightest change in one of the parameters can result in drastic changes in the outcome and this makes it difficult to carry out such predictions. There have been many attempts to predict the outcome of the football match but the uncertainty of certain aspects makes it difficult. Perhaps one of the major reasons for the love of the sport is because of the fact that even though all the odds are pitted against, sheer will, hope and hard work can change everything. One such case occurred when Leicester football which was one of the most underrated in the English premier league shocked everyone with their performance and went on to become the champions of the 2015 English premier league beating even the best of the teams which were thought to be unbeatable.These kind of events are also what makes this sport so much more beautiful and entertaining for all the fans This report aims to utilize football data to try and apply machine learning algorithms and get a better understanding of the machine learning process. The report tries to formulate a solution for the prediction and classification problems in our football dataset.

A. How many goals will a team score by the end of the match?

Football is a very tricky sport where the game can change in favor of any team at any moment of time. Even a small event can cause drastically change the expected result within seconds. The most dominating team in the game can be shocked by a sudden comeback at any given time in the game and increase the uncertainty till the blow of the final whistle.Prediction of the number of goals in a football match depends on a lot of variables. The factors such as the location of a match (i.e. home or away), opponent performance, number of fouls, etc. can affect the game at any point. This report will analyze various football stats and data along with the half-time score and the probabilities predicted by popular betting sites to predict the final scores of the match.

B. What will be the match result for the given team?

The prediction of match outcome can be utilized by individuals as well as a team to get the overall performance of the team and to create strategies in advance. There is a lot of difference in a team's performance when they are playing on the home ground compared to away ground. This report takes into consideration all the valuable factors that can affect the outcome of a match and tries to predict the result of the game (i.e. win, loss and draw). We use classification to classify the match result of whether it is a win, loss, or draw based on the data.

II. Related Work

- A. Azeman, A. Mustapha, N. Razali, A. Nanthaamomphong and M. H. Abd Wahab studied the outcome of football matches for the English premier league. The prediction was done to classify the outcome of the match into three classes i.e. win, lose or draw. They used multiclass decision Forest and neural network models for the task. The accuracy for the decision forest was the highest. However, the dataset used is only for one league and one season which makes it a very small dataset and reduces the variety. [1]
- O. Hubáček, G. Šourek, and F. Železný experimented with prediction of future matches .They computed pi-rating and a rating based on the PageRank method. They performed a feature based classification model. They used the Xgboost model for thier task. They discussed how thorough analysis of individual constructed features can improve performance. [2]
- E. Tiwari, P. Sardar and S. Jain made the prediction of football match results using Recurrent Neural networks and the LSTM model. They found out that RNN with LSTM performed very well with the best accuracy of 68%. They discussed how the model can be used for different sports classification problems as well such as the number of goals prediction [3]
- M. S. Oughali, M. Bahloul and S. A. El Rahman did a comparative study to predict shooting success for basketball. They used Random forest and Xgboost for shooting predictions. They found out that the Xgboost model performed well and had the highest accuracy of 60%. They discussed how the sports predictions are tricky and the results are slightly off-balance due to the human factor being involved. [4]
- T. Korotyeyeva, R. Tushnytskyy and V. Kulyk studied the forecasting of football matches using neural networks. They were able to achieve 75% accuracy with the neural network. The accuracy of this model can be increased using more hidden layers in the neural network and using data from different leagues. This will helpthe model to be overall accurate and not be specific to just one league of football. [5]
- • D. Prasetio and D. Harlili used Logistic regression using variables like “Home Offense”, “Home Defense”, “Away Offense”, and “Away Defense”.The model was able to achieve an accuracy of 69.5% accuracy. They used 2015-2016 Barclays’ Premier League data which is comparatively a smaller dataset for this prediction. The features that were calculated can be improved by getting real time data to tweak the features that were created for the purpose of regression. [6]
- • C. Pipatchatchawal and S. Phimoltares used video game ratings of players and teams to utilize fusion-based models. They used a hierarchical model and ensemble model for prediction on the English Premier League dataset. They were able to achieve the highest accuracy of about 56.80%. Adding feature analysis and selection can make the training process much more efficient in this task. [7]
- J. Hucaljuk and A. Rakipović predicted the match outcome of the champion Champions League matches. To achieve a good outcome on the results they tested the dataset with many classifiers. They were able to achieve 65% accuracy. This model can be improved by improving feature selection and working on a larger dataset.The improvement in feature selection will lead to improved overall performance of the model. [8]
- X. Tang, Z. Liu, T. Li, W. Wu and Z. Wei applied a decision tree for the classification of the winning team in CFASL. They were able to achieve an accuracy of 57.7% in their model. This accuracy can be increased by using the Random forest algorithm on this dataset.The accuracy can be increased further by implementing Xgboost methodology which is an ensemble based model. [9]
- K. Huang and W. Chang used the supervised multi-layer perceptron neural network along with error backpropagation for the purpose of predicting the winning rate of a team for the 2006 World Cup. After the exclusion of tied games, the accuracy achieved was 76.9%. This model works poorly when tied games are considered. The football games that end up in a draw are really important to be considered and should not be ignored while building a model. [10]
- Tianxiang Cui, Jingpeng Li, J. R. Woodward and A. J. Parkes, used an ensemble-based Genetic Programming system for the purpose of predicting the English Premier League. The functions generated by genetic programming were able to achieve an accuracy of 70%. They used 25 features for prediction. [11]
- L. Hervert-Escobar, T. I. Matis and N. Hernandez used rank position-based Bayesian model and historical data for predicting the outcome of FIFA world cup 2018. They were able to predict the teams moving to the next stage with 69% accuracy. Their future work includes predicting the accurate number of goals in a match.They can use the data from this model to create a performance parameter and apply LSTM to find a team’s current form which will help in prediction of number of goals. [12]
- S. Dobravec used latent features to build a model for FIFA World Cup 2014 tournament and used the cross fold method to evaluate the model. The major problem is that the use of short-termed datasets with only 64 matches data. [13]
- N. Danisik, P. Lacko and M. Farkas made use of player data to predict the outcome of a match which might be home team victory, away team victory or draw. They used LSTM and were able to predict with the accuracy of 52.47%. The model will work better with some more of the attributes of the player to build a more reliable model. [14]

III. Methodology

For this report, we are going to use the Knowledge Discover in Database or KDD methodology is a process of extracting knowledge from data. The KDD methodology is suitable for us to research our machine learning algorithms and draw suitable conclusions.Fig 1 depicts the flow of KDD process

- In the first step we understand the domain and application of data and try to determine the end goal .
- In data selection process we try to find a dataset that fulfills our requirements to achieve our goals.
- After selecting our data we perform data cleaning and preprocessing where we remove all the outliers, null values and unwanted noise from our data
- In data reduction process we select the useful feature of the data and perform dimensionality reduction and transformation to make our data much more effective
- In this process we determine whether the goal is classification, regression or clustering and we select appropriate algorithms for our task.
- The algorithms are implemented on the data and we use various technics to make the algorithm mor effective.
- We select the best algorithm which performs better for our task.
- We draw conclusion from the consolidated discovered knowledge.

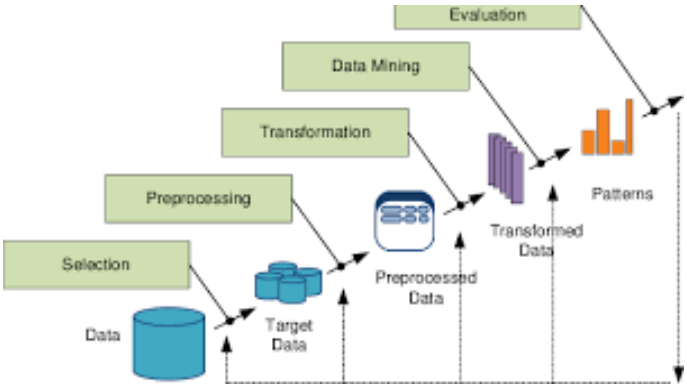

Fig. 1. KDD process

IV. Implementation of Models

This report makes use of a football dataset from Kaggle which has sevenfiles with data arranged in a relational manner. We will be using data by getting the required elements from these files to merge to make a dataset that will help us achieve our task. For our first research question, we will be using two similar datasets which have the major difference of football leagues associated with the data and the data of different betting platforms along with halftime score data. The second research question will utilize one dataset for the classification of the match result

A. Dataset 1

1) : For our first dataset, we will be merging two datasets with the gameID column, and we also filtered the data we used data from EPL, Serie A and Bundesliga leagues data from Understat and bet365 platforms. The dataset was cleaned by removing all the unnecessary columns and cleaning four NA values by dropping the rows. The teams which had the lowest number of matches in the dataset were also dropped to allow proper training on the relevant dataset and a cleaned dataset was prepared. We transformed the three categorical variables into factors for our model training and made our data ready for applying machine learning algorithms. Fig 2 shows the summary of the data that will be used.

```
> summary(finalDataGoals)
     teamID          season          location      goals            xGoals           shots       shotsOnTarget        deep             ppda
 72     : 266   2014:2280   a:7978   Min.   : 0.000   Min.   :0.0000   Min.   : 0.00   Min.   : 0.000   Min.   : 0.000   Min.   : 2.250
 74     : 266   2015:2280   h:7978   1st Qu.: 0.000   1st Qu.:0.7035   1st Qu.: 9.00   1st Qu.: 3.000   1st Qu.: 3.000   1st Qu.: 7.078
 75     : 266   2016:2280            Median : 1.000   Median :1.1867   Median :12.00   Median : 4.000   Median : 5.000   Median : 9.632
 78     : 266   2017:2280            Mean   : 1.363   Mean   :1.3429   Mean   :12.52   Mean   : 4.291   Mean   : 6.086   Mean   :11.063
 80     : 266   2018:2280            3rd Qu.: 2.000   3rd Qu.:1.8183   3rd Qu.:16.00   3rd Qu.: 6.000   3rd Qu.: 8.000   3rd Qu.:13.274
 81     : 266   2019:2280            Max.   :10.000   Max.   :6.6305   Max.   :47.00   Max.   :18.000   Max.   :42.000   Max.   :97.333
 (Other):14360   2020:2276
     fouls          corners        yellowCards       redCards       homeGoalsHalfTime awayGoalsHalfTime     B365H            B365D
 Min.   : 0.00   Min.   : 0.000   Min.   : 1.000   Min.   :0.0000   Min.   :0.0000    Min.   :0.0000    Min.   : 1.020   Min.   : 2.400
 1st Qu.:10.00   1st Qu.: 3.000   1st Qu.: 2.000   1st Qu.:0.0000   1st Qu.:0.0000    1st Qu.:0.0000    1st Qu.: 1.650   1st Qu.: 3.300
 Median :13.00   Median : 5.000   Median : 3.000   Median :0.0000   Median :0.0000    Median :0.0000    Median : 2.200   Median : 3.600
 Mean   :12.83   Mean   : 5.087   Mean   : 3.153   Mean   :0.1032   Mean   :0.6732    Mean   :0.5283    Mean   : 2.882   Mean   : 4.146
 3rd Qu.:16.00   3rd Qu.: 7.000   3rd Qu.: 4.000   3rd Qu.:0.0000   3rd Qu.:1.0000    3rd Qu.:1.0000    3rd Qu.: 3.200   3rd Qu.: 4.330
 Max.   :33.00   Max.   :20.000   Max.   :11.000   Max.   :3.0000   Max.   :5.0000    Max.   :5.0000    Max.   :26.000   Max.   :17.000

     B365A
 Min.   : 1.08
 1st Qu.: 2.30
 Median : 3.40
 Mean   : 4.90
 3rd Qu.: 5.50
 Max.   :41.00
```

Fig. 2. summary of Dataset 1

2) : For model selection, we select multiple linear regression as it is very easy to implement the model and effective and easy to understand for regression problem we implemented this model. We have selected goals as our dependent variable and for the model building we have plotted the correlation as shown in Fig. 3 for all the variables and found out that two columns B365A and B365H have a very high correlation

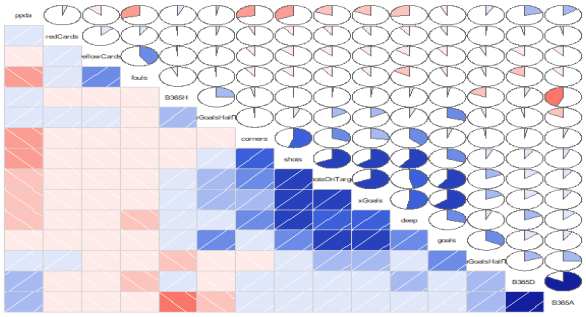

Fig. 3. Correlation for Dataset 1

3) : We decide to remove B365H to fix this problem for training our multiple linear regression without overfitting. After taking a look at our dependent variable we see that the data starts from 0 which is accurate as per the football matches scenario as can be seen in Fig. 4.

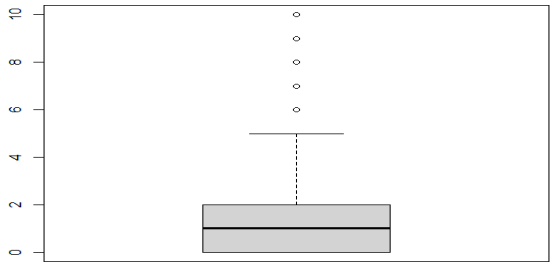

Fig. 4. Boxplot of dependent variable

4) : For selecting the features for our model we have performed Boruta's algorithm to check for the importance of all the variables for the prediction of our target variable 'goals'. Boruta's algorithm returns all the variables as important for prediction as depicted in Fig. 5

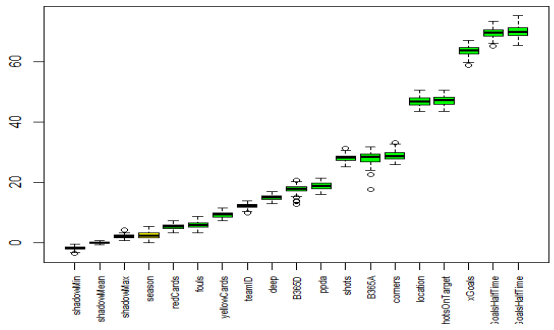

Fig. 5. Important variables for Dataset 1

5) : We first split our data into train and test split and used 70 percent of data for training and the rest for testing. Now we built our first model with all the variables in our multiple linear regression and we get the results as shown below in Fig. 6 with an accuracy of 59.75%

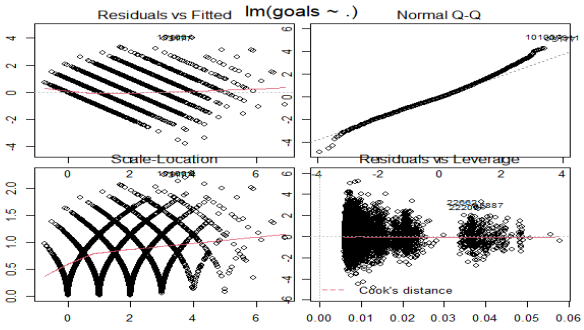

Fig. 6. First multiple linear regression model

6) : For model improvements as we can see there are a lot of outliers that are influencing our predictions. To improve the performance of our model we will remove all the outliers with Cook's distance greater than 1 and recompute our model we also remove the variables that became insignificant while training and running the model.

7) : After we have found a satisfying model as shown in Fig.Fig. 7 we checked whether it follows the multiple linear regression assumptions aby carry out Durbin Watson and VIF tests and we found that our model satisfies the assumptions as shown by the Fig. 8. For evaluation of our model, we tested our model on the test dataset and got RMSE as 0.78, RSquare as 0.60 and MAPE as 0.77 from our multiple linear regression model.

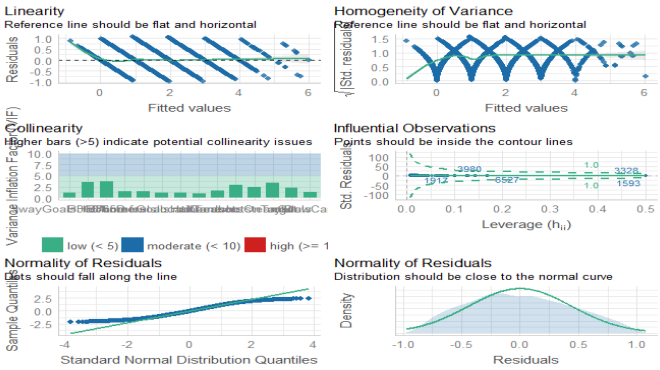

Fig. 7. Final model for multiple linear regression

```
> durbinWatsonTest(lm.fit1)
 lag Autocorrelation D-W Statistic p-value
   1     0.002787366      1.994015   0.848
 Alternative hypothesis: rho != 0
> vif(lm.fit1)
                     GVIF Df GVIF^(1/(2*Df))
teamID           4.530649 91        1.008336
season           1.713126  6        1.045881
location         1.104411  1        1.050910
xGoals           2.374223  1        1.540851
shots            3.116870  1        1.765466
shotsOnTarget    2.422689  1        1.556499
deep             2.024372  1        1.422804
ppda             1.652279  1        1.285410
fouls            1.616778  1        1.271526
corners          1.524492  1        1.234703
yellowCards      1.312383  1        1.145593
redCards         1.054172  1        1.026729
homeGoalsHalfTime 1.135705  1        1.065695
awayGoalsHalfTime 1.124324  1        1.060341
B365D            3.897637  1        1.974243
B365A            3.523068  1        1.876984
```

Fig. 8. Evaluation of assumptions for multiple linear regression

8) : For evaluation of our model we tested our model on test dataset and got RMSE as 0.78 ,RSquare as 0.60 and MAPE as 0.77 from our multiple linear regression model.

```
> eval_results(y.test, predictions_test,test)
      RMSE   Rsquare      MAPE
1 0.7869715 0.6099926 0.770737
```

Fig. 9. Evaluation of model

B. Dataset 2

1) : For our second dataset, we used the data from Bundesliga, Ligue 1 and La Liga for football leagues and Betway and William Hill betting Platforms. We cleaned the NA values by dropping them and creating a summary of our dataset as shown in Fig. 10.

```
> summary(finalDataGoals)
     teamID        season      location      goals            xGoals            shots       shotsOnTarget        deep       
 138    :  266   2014:2132   a:7357   Min.   : 0.000   Min.   :0.0000   Min.   :  0.00   Min.   : 0.000   Min.   :  0.00  
 140    :  266   2015:2132   h:7357   1st Qu.: 0.000   1st Qu.:0.6897   1st Qu.:  9.00   1st Qu.: 3.000   1st Qu.:  3.00  
 143    :  266   2016:2128            Median : 1.000   Median :1.1665   Median : 12.00   Median : 4.000   Median :  5.00  
 146    :  266   2017:2132            Mean   : 1.362   Mean   :1.3305   Mean   : 12.13   Mean   : 4.286   Mean   :  5.45  
 147    :  266   2018:2132            3rd Qu.: 2.000   3rd Qu.:1.7950   3rd Qu.: 15.00   3rd Qu.: 6.000   3rd Qu.:  7.00  
 148    :  266   2019:1930            Max.   :10.000   Max.   :6.6109   Max.   : 37.00   Max.   :17.000   Max.   : 42.00  
 (other):13118   2020:2128                                                                                                  
      ppda            fouls           corners        yellowCards       redCards        homeTeamID      awayTeamID   
 Min.   :  1.897   Min.   : 0.00   Min.   : 0.000   Min.   :0.000   Min.   :0.0000   138    :  266   138    :  266  
 1st Qu.:  6.883   1st Qu.:10.00   1st Qu.: 3.000   1st Qu.:1.000   1st Qu.:0.0000   140    :  266   140    :  266  
 Median :  9.429   Median :13.00   Median : 4.000   Median :2.000   Median :0.0000   143    :  266   143    :  266  
 Mean   : 10.799   Mean   :13.37   Mean   : 4.794   Mean   :2.079   Mean   :0.1072   146    :  266   146    :  266  
 3rd Qu.: 12.960   3rd Qu.:16.00   3rd Qu.: 6.000   3rd Qu.:3.000   3rd Qu.:0.0000   147    :  266   147    :  266  
 Max.   :152.000   Max.   :33.00   Max.   :20.000   Max.   :9.000   Max.   :3.0000   148    :  266   148    :  266  
                                                                                     (other):13118   (other):13118  
 homeGoalsHalftime awayGoalsHalftime      BWH              BWD              BWA              WHH              WHD        
 Min.   :0.0000    Min.   :0.0000    Min.   : 1.030   Min.   : 2.050   Min.   :  1.02   Min.   : 1.030   Min.   : 2.200  
 1st Qu.:0.0000    1st Qu.:0.0000    1st Qu.: 1.720   1st Qu.: 3.250   1st Qu.:  1.73   1st Qu.: 1.720   1st Qu.: 3.200  
 Median :0.0000    Median :0.0000    Median : 2.200   Median : 3.500   Median :  2.20   Median : 2.200   Median : 3.500  
 Mean   :0.6709    Mean   :0.5137    Mean   : 2.737   Mean   : 4.003   Mean   :  2.76   Mean   : 2.737   Mean   : 3.914  
 3rd Qu.:1.0000    3rd Qu.:1.0000    3rd Qu.: 2.850   3rd Qu.: 4.100   3rd Qu.:  2.88   3rd Qu.: 2.850   3rd Qu.: 4.000  
 Max.   :6.0000    Max.   :5.0000    Max.   :34.000   Max.   :19.500   Max.   : 26.00   Max.   :34.000   Max.   :19.000  

      WHA       
 Min.   : 1.08  
 1st Qu.: 2.50  
 Median : 3.40  
 Mean   : 4.70  
 3rd Qu.: 5.00  
 Max.   :51.00  
```

Fig. 10. Summary of Dataset 2

2) : For Feature selection, we used Boruta's algorithm to determine the importance of all variables for the model building process. We found all the variables to be important for the model building process so we will be using them all. The Boruta algorithm also shows that awayGoalsHalftime, xGoals and homeGoalsHalftime are the most important variables as seen in Fig. 11.

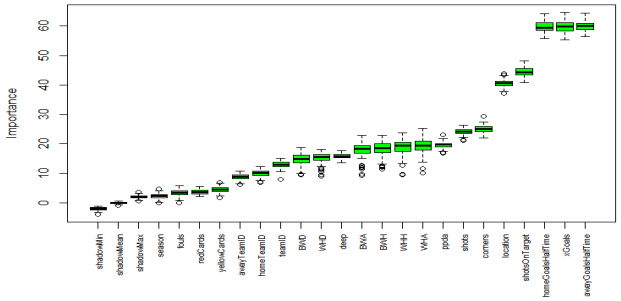

Fig. 11. Feature Selection using Boruta's algorithm

3) : We will be using Ridge and Lasso Regression for our second dataset as these two models reduce model complexity and are good for avoiding over fitting. For ridge regression we used glmnet Library and we set the alpha as 0 . We can improve the model's performance by tuning the values of Lambda. We are going to use grid search method to evaluate the values of Lamda for our model which we can refer in Fig. 12.

4) : After applying grid search we arrived at an optimal lambda of 0.001 and we can use this lambda for finding the optimal ridge regression model for our data. After looking at the L1 norm vs coefficient plot it is clear that we get a perfect

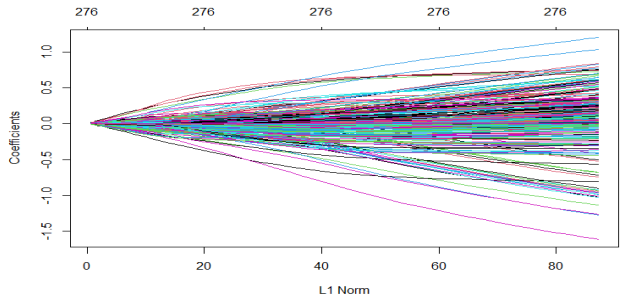

Fig. 12. Lambda parameter tuning

model when our lambda is close to zero as shown in the plot Fig. 13.

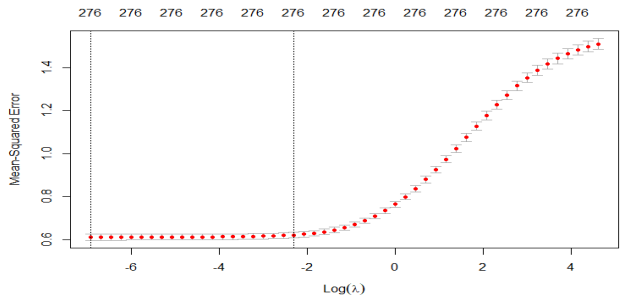

Fig. 13. L1 VS Coefficient plot

5) : Evaluating this model on our test dataset we get the RMSE of 0.79, RSquare of 0.61 and MAPE of 1.507.

```
> eval_results(y.test, predictions_test, test)
      RMSE  Rsquare     MAPE
1 0.7928073 0.613878 1.507903
```

Fig. 14. Evaluating Ridge regression

6) : For Lasso regression, we will set the alpha of glmnet as 1 and try to evaluate our model based on the same parameters. We will also create a grid to hyper-tune lambda parameters and arrive at an optimal solution shown in Fig. 15.

7) : We arrived at lambda 0.006 as the optimal solution and used the parameter in our lasso regression model to evaluate our test data and we got RMSE as 0.79, RSquare as 0.61 and MAPE as 0.84

8) : After evaluating both the models we can see that Ridge regression is better suited for our task as it gives the highest accuracy and also it has a lower RMSE score. Thus, Ridge regression performed the best for our task to predict the number of goals scored in a match. This might not be a perfect result and can be improved upon in the future as the data does not show individual player performance which can drastically change the prediction accuracy of our model.

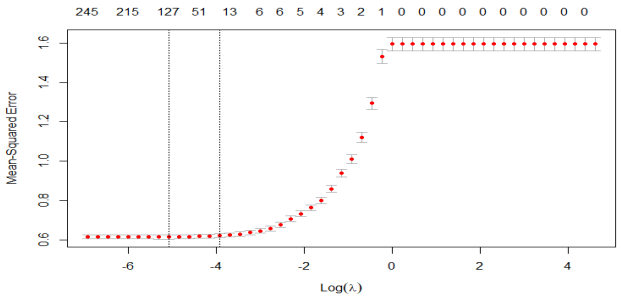

Fig. 15. Lambda tuning for Lasso

```
> eval_results(y.test, predictions_test, test)
      RMSE    Rsquare      MAPE
1 0.7933298 0.6133689 0.8497257
> |
```

Fig. 16. Evaluating Lasso regression

C. Dataset 3

For our classification task, we are going to use data by combining games and team stats data and by working on the result as our target variable which has 3 values W, L and D (i.e. Win, Lose and draw respectively. We have cleaned the data and dropped all the NA values and unwanted columns. The Summary for our data is shown in Fig. 17.

```
> summary(finalData)
     teamID          season     location     xGoals           shots       shotsOnTarget        deep             ppda
 72     : 258   2014:1367   a:4942   Min.   :0.0000   Min.   : 0.00   Min.   : 0.000   Min.   : 0.000   Min.   :  2.441
 74     : 258   2015:1444   h:4941   1st Qu.:0.7157   1st Qu.: 9.00   1st Qu.: 3.000   1st Qu.: 3.000   1st Qu.:  7.500
 75     : 258   2016:1368            Median :1.2005   Median :12.00   Median : 4.000   Median : 5.000   Median : 10.152
 78     : 258   2017:1520            Mean   :1.3457   Mean   :12.89   Mean   : 4.345   Mean   : 6.318   Mean   : 11.735
 80     : 258   2018:1444            3rd Qu.:1.8307   3rd Qu.:16.00   3rd Qu.: 6.000   3rd Qu.: 8.000   3rd Qu.: 14.046
 81     : 258   2019:1372            Max.   :6.6305   Max.   :47.00   Max.   :18.000   Max.   :37.000   Max.   :97.333
 (Other):8335   2020:1368
     fouls           corners        yellowCards      redCards       result    homeProbability   drawProbability   awayProbability
 Min.   : 0.0   Min.   : 0.000   Min.   :0.000   Min.   :0.00000   D:2419   Min.   :0.0000   Min.   :0.0003   Min.   :0.0000
 1st Qu.: 9.0   1st Qu.: 3.000   1st Qu.:1.000   1st Qu.:0.00000   L:3732   1st Qu.:0.1855   1st Qu.:0.1623   1st Qu.:0.0923
 Median :12.0   Median : 5.000   Median :2.000   Median :0.00000   W:3732   Median :0.4250   Median :0.2460   Median :0.2543
 Mean   :12.3   Mean   : 5.232   Mean   :1.959   Mean   :0.09744            Mean   :0.4416   Mean   :0.2349   Mean   :0.3235
 3rd Qu.:15.0   3rd Qu.: 7.000   3rd Qu.:3.000   3rd Qu.:0.00000            3rd Qu.:0.6746   3rd Qu.:0.3039   3rd Qu.:0.5114
 Max.   :32.0   Max.   :20.000   Max.   :9.000   Max.   :2.00000            Max.   :0.9996   Max.   :0.8439   Max.   :0.9997

    opponent
 72     : 258
 74     : 258
 75     : 258
 78     : 258
 80     : 258
 81     : 258
 (Other):8335
```

Fig. 17. Summary of Dataset 3

1) : To determine which of these variables are important to us and which we can drop out of the model to make our model more effective is necessary. To do this we use Boruta's algorithm. The algorithm shows that all the variables we have selected are important as depicted in Fig.Fig. 18. Therefore, we decide to keep all the variables for training our model.

2) : For this classification problem we will be using Random forest and Xgboost as per the research paper we referred . [4] This type of classification works a lot better as we saw in our research review and we will be implementing the same for excellent results.

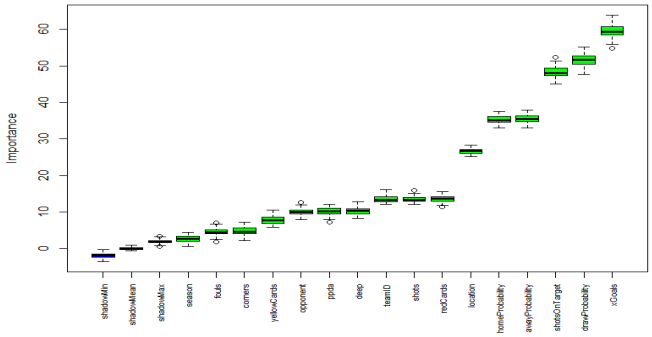

Fig. 18. Feature Selection using Boruta's algorithm

3) : First, we applied random forest for the classification problem the random forest is an ensemble learning method that works better than decision trees for classification. But just running random forest is not enough we will be using hyperparameter tuning to get better results. We will be calculating the optimal mtry value which is the number of variables selected at each split.

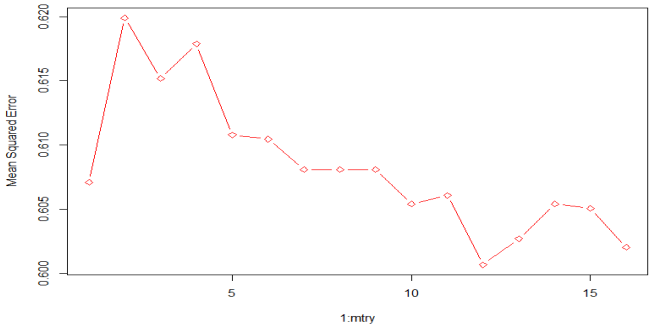

Fig. 19. Mtry parameter tuning

4) : As the plot in Fig. 19 shows the mtry with value 2 gives the model the best accuracy. We will apply this value to our model and evaluate our model on the testing dataset. On evaluating the model we get an accuracy of 0.618 and a Kappa value of 0.399.

5) : Now we will be using XGBoos which stands for extreme gradient boosting. This is a better version of the gradient boost model and performs better than traditional gradient boosting most of the time. We will be using this model and will be hyper tuning to increase our accuracy and find the best parameters with the help of trainControl which will help us find the best tuning parameters for our data as depicted in Fig. 21.

6) : The best tuning parameters will be applied to our model to build a better model for our dataset to calculate the value of the result. The Xgboost gives an accuracy of 0.610 with a Kapa of 0.41.

```
> confusionMatrix(predopt,y.test)
Confusion Matrix and Statistics

          Reference
Prediction   D   L   W
         D 117  80  86
         L 307 870 160
         W 305 193 847

Overall Statistics

               Accuracy : 0.6185
                 95% CI : (0.6008, 0.6361)
    No Information Rate : 0.3855
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3997

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: D Class: L Class: W
Sensitivity           0.16049   0.7612   0.7749
Specificity           0.92576   0.7437   0.7340
Pos Pred Value        0.41343   0.6507   0.6297
Neg Pred Value        0.77181   0.8323   0.8481
Prevalence            0.24587   0.3855   0.3686
Detection Rate        0.03946   0.2934   0.2857
Detection Prevalence  0.09545   0.4509   0.4536
Balanced Accuracy     0.54313   0.7524   0.7545
>
```

Fig. 20. Random forest model evaluation

```
> model$bestTune
   nrounds max_depth eta gamma colsample_bytree min_child_weight subsample
90     150         2 0.4     0              0.8                1         1
```

Fig. 21. Hyper parameter tuning result for Xgboost

7) : For evaluation of the best model for our task, we will look at accuracy as well as Kappa value. The accuracy tells us the correctness of our model while the Kappa gives us the reliability. In this case, even if the accuracy score achieved by both Random Forest and Xgboost are the same which is 0.61, the Kapa score of Xgboost is 0.41 which is higher than that of random forest which is 0.39. Therefore, Xgboost performed better for this task.

V. Conclusion

With proper understanding, research and learning we were able to implement the machine learning methodologies to answer our research questions. The regression problem of predicting the number of goals scored by a team was addressed with multiple linear regression, Ridge regression and Laso regression. The ridge regression performed best, and we were able to achieve accuracy up to 61%. The classification problem for classifying the match result as a win, lose or draw was tackled using Random Forest and Xgboost models and we were able to build a model with Xgboost which had the highest accuracy of 61%. Considering sports has a lot of factors involved and the difficulty to properly predict. The underlying human physical and mental efforts put into the game by players are difficult to quantify which makes 61% accuracy pretty good for research questions.

```
> confusionMatrix(predicted.classes,y.test)
Confusion Matrix and Statistics

          Reference
Prediction   D   L   W
         D 167 124 107
         L 281 848 139
         W 281 171 847

Overall Statistics

               Accuracy : 0.628
                 95% CI : (0.6103, 0.6454)
    No Information Rate : 0.3855
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4193

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: D Class: L Class: W
Sensitivity           0.22908   0.7419   0.7749
Specificity           0.89669   0.7695   0.7585
Pos Pred Value        0.41960   0.6688   0.6520
Neg Pred Value        0.78107   0.8262   0.8523
Prevalence            0.24587   0.3855   0.3686
Detection Rate        0.05632   0.2860   0.2857
Detection Prevalence  0.13423   0.4277   0.4381
Balanced Accuracy     0.56289   0.7557   0.7667
>
```

Fig. 22. Xgboost model evaluation

VI. Future Work

The game of football highly depends on the players and their teamwork and having the data for the overall performance of each player in the team as well as the overall team strength can make the prediction much more accurate. The use of neural networks for the prediction can also be explored to increase the accuracy of our model in the future.

References

[1] A. A. Azeman, A. Mustapha, N. Razali, A. Nanthaamomphong and M. H. Abd Wahab, "Prediction of Football Matches Results: Decision Forest against Neural Networks," 2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2021, pp. 1032-1035, doi: 10.1109/ECTI-CON51831.2021.9454789.

[2] Hubáček, Ondřej, Gustav Šourek, and Filip Železný. "Learning to predict soccer results from relational data with gradient boosted trees." Machine Learning 108, no. 1 (2019): 29-47.

[3] Tiwari, P. Sardar and S. Jain, "Football Match Result Prediction Using Neural Networks and Deep Learning," 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2020, pp. 229-231, doi: 10.1109/ICRITO48877.2020.9197811.

[4] M. S. Oughali, M. Bahloul and S. A. El Rahman, "Analysis of NBA Players and Shot Prediction Using Random Forest and XGBoost Models," 2019 International Conference on Computer and Information Sciences (ICCIS), 2019, pp. 1-5, doi: 10.1109/ICCISci.2019.8716412.

[5] T. Korotyeyeva, R. Tushnytskyy and V. Kulyk, "Applying Neural Networks to Football Matches Results Forecasting," 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), 2018, pp. 278-282, doi: 10.1109/STC-CSIT.2018.8526756.

[6] D. Prasetio and D. Harlili, "Predicting football match results with logistic regression," 2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA), 2016, pp. 1-5, doi: 10.1109/ICAICTA.2016.7803111.

[7] C. Pipatchatchawal and S. Phimoltares, "Predicting Football Match Result Using Fusion-based Classification Models," 2021 18th International Joint Conference on Computer Science and Software Engineering (JCSSE), 2021, pp. 1-6, doi: 10.1109/JCSSE53117.2021.9493837.

[8] J. Hucaljuk and A. Rakipović, "Predicting football scores using machine learning techniques," 2011 Proceedings of the 34th International Convention MIPRO, 2011, pp. 1623-1627.

[9] X. Tang, Z. Liu, T. Li, W. Wu and Z. Wei, "The Application of Decision Tree in the Prediction of Winning Team," 2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), 2018, pp. 239-242, doi: 10.1109/ICVRIS.2018.00065.

[10] K. Huang and W. Chang, "A neural network method for prediction of 2006 World Cup Football Game," The 2010 International Joint Conference on Neural Networks (IJCNN), 2010, pp. 1-8, doi: 10.1109/IJCNN.2010.5596458.

[11] Tianxiang Cui, Jingpeng Li, J. R. Woodward and A. J. Parkes, "An ensemble based Genetic Programming system to predict English football premier league games," 2013 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), 2013, pp. 138-143, doi: 10.1109/EAIS.2013.6604116.

[12] L. Hervert-Escobar, T. I. Matis and N. Hernandez-Gress, "Prediction Learning Model for Soccer Matches Outcomes," 2018 Seventeenth Mexican International Conference on Artificial Intelligence (MICAI), 2018, pp. 63-69, doi: 10.1109/MICAI46078.2018.00018.

[13] S. Dobravec, "Predicting sports results using latent features: A case study," 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2015, pp. 1267-1272, doi: 10.1109/MIPRO.2015.7160470.

[14] N. Danisik, P. Lacko and M. Farkas, "Football Match Prediction Using Players Attributes," 2018 World Symposium on Digital Intelligence for Systems and Machines (DISA), 2018, pp. 201-206, doi: 10.1109/DISA.2018.8490613.